ITGS301: TUTORIAL

Mai Elbaabaa

REVIEW

- Master theorem method
- Recurrence tree method
- Kruscl Vs Prime’s Algorithms
- 2 Questions from midterm

MASTER THEOREM

- The Master Method is used for solving the following types of recurrence
- $T(n) = a\,T(n/b) + f(n)$
- Where a=> 1, b>1, and f is a function, f(n) > 0.
  - *Idea: compare f(n) with* $n^{\log_b a}$
- Case 1: $T(n) = \Theta(n^{\log_b a})$ if $f(n) < n^{\log_b a}$
- Case 2: $T(n) = \Theta(n^{\log_b a} \lg n)$ if $f(n) = n^{\log_b a}$
- Case 3: $T(n) = \Theta(f(n))$ if $f(n) > n^{\log_b a}$

$T(n) = 3T(n/2) + n^2$

Here,

a = 3

n/b = n/2

$f(n) = n^2$

$\log_b a = \log_2 3 \approx 1.58 < 2$

ie. $f(n) < n^{\log b\ a}$.

Case 3 implies here.

Thus, $T(n) = f(n) = \Theta(n^2)$

$$y = \log_b x \quad \Rightarrow \quad b^y = x$$

$$b^{\log_b x} = x$$

The master theorem cannot be used if:

T(n) is not monotone. eg. T(n) = sin n
f(n) is not a polynomial. eg. f(n) = 2^n
a is not a constant. eg. a = 2n
a < 1

$$T(n) = a\,T\left(\frac{n}{b}\right) + \theta\left(n^k \log^p n\right)$$

Master's Theorem

1. if $a > b^k$, then $T(n) = \theta(n^{\log_b a})$
2. if $a = b^k$, then
   (a) if $p > -1$, then $T(n) = \theta(n^{\log_b a} \log^{p+1} n)$
   (b) if $p = -1$, then $T(n) = \theta(n^{\log_b a} \log\log n)$
   (c) if $p < -1$, then $T(n) = \theta(n^{\log_b a})$
3. if $a < b^k$, then
   (a) if $p >= 0$, then $T(n) = \theta(n^k \log^p n)$
   (b) if $p < 0$, then $T(n) = \theta(n^k)$

- Here, a >= 1, b > 1, k >= 0 and p is a real number.
- T (n) = 6T (n/3) + n^2log n
- a =6, b=3, k=2 , P=1
- a b^k
- 6 3^2
- 6 < 9 ,case 3 p >=0
- T (n) = $\Theta(n^2 \log n)$ (Case 3)

Master's Theorem can only be used for recurrence relations of the form:
•T(n)$T(n)$ = aT(n/b) + f(n), where $f(n)=\theta(n^k log^p n)$ (Dividing Recurrence Relation)

MASTER THEOREM EXAMPLES

1. T (n) = 16T (n/4) + n! =⇒ T (n) = Θ(n!) (Case 3)
2. T (n) = √2T (n/2) + log n =⇒ T (n) = Θ(√n) (Case 1)
3. T (n) = 3T (n/2) + n =⇒ T (n) = Θ(n^lg 3) (Case 1)
4. T (n) = 3T (n/3) + √n =⇒ T (n) = Θ(n) (Case 1)
5. T (n) = 2T (n/2) + n/ log n =⇒ Does not apply (f(n)) ⇒- nlogn negative

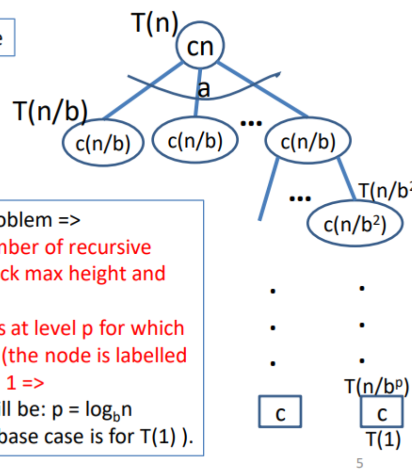
RECURRENCE TREE METHOD

T(1) = c

Problem size

The local TC at the node

$$T(n) = a * T(n/b) + cn$$



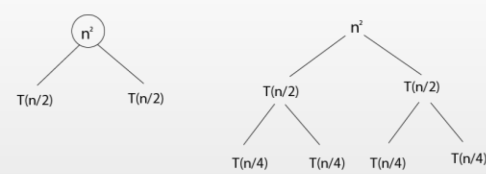
Number of subproblems => Number of children of a node in the recursion tree. => Affects the number of nodes per level. At level i there will be a^i nodes.
Affects the level TC.

Size of a subproblem =>
Affects the number of recursive calls (frame stack max height and tree height)
Recursion stops at level p for which the pb size is 1 (the node is labelled T(1)) => $n/b^p = 1$ =>
Last level, p, will be: $p = \log_b n$
(assuming the base case is for T(1)).

TC = time complexity

5

- solve the following equation using recurrence tree method
- $T(n) = 2T(n/2) + n^2$

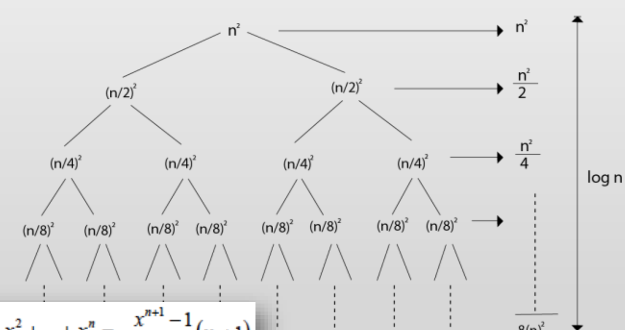


$$T(n) = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \ldots\ldots \log n \text{ times.}$$

$$\leq n^2 \sum_{i=0}^{\infty} \left(\frac{1}{2^i}\right)$$

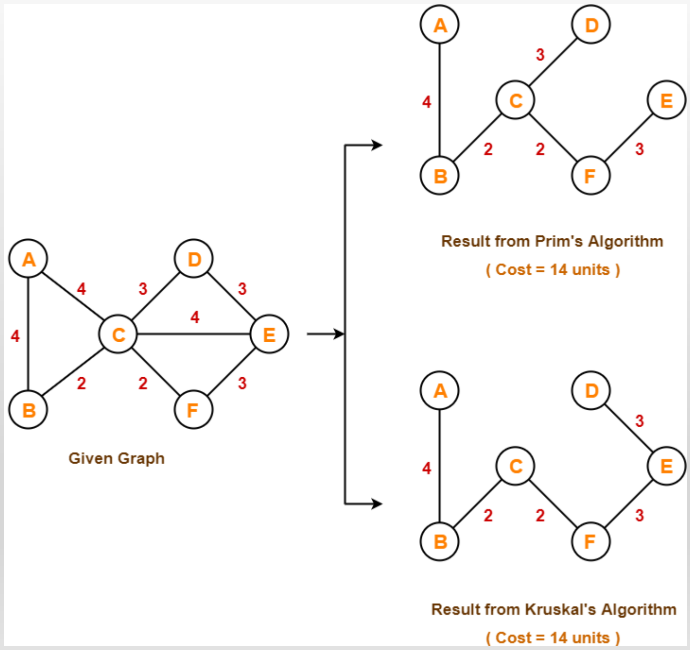
$$\leq n^2 \left(\frac{1}{1-\frac{1}{2}}\right) \leq 2n^2$$

$$\mathbf{T(n) = \theta n^2}$$



- **Geometric series:** $\sum_{k=0}^{n} x^k = 1 + x + x^2 + \ldots + x^n = \frac{x^{n+1} - 1}{x - 1} (x \neq 1)$
  - **Special case: x < 1:** $\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$

PRIM’S AND KRUSKAL’S ALGORITHMS-



QUESTION FROM MIDTERM

- Big oh of *f* (*n*) = *4 + 8 + 12 + … + 4n*
- F(n)= 4(1+2+3+4…..+n)
- F(n)= 4n(n+1)/2
- F(n)= $2n^2$+2
- O(f(n))=O(n^2)

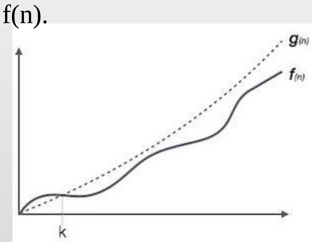
- **Arithmetic series:** $\sum_{k=1}^{n} k = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$

QUESTION FROM MIDTERM

- asymptotic notations
- F(n) = 6$\boldsymbol{n^4}$ + 4$\boldsymbol{n^2}$ + log n
- T(n) = $\Omega(n^2)$ lower bound
- T(n) = O (n^4)
- T(n) = Θ (n^4)
- T(n)=O(n^2) upper bound worst case false

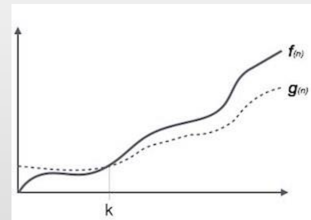
Big Oh Notation, O

This implies that f(n) does not grow faster than g(n), or g(n) is an upper bound on the function f(n).



Omega Notation, Ω

It is used to bound the growth of running time for large input size. f(n) is on the order of g(n))



Assume T(n)=c1n^2+c2n for c1 and c2>0. Then,

C1 n^2+c2 n≥ c1 n2
for all n>1. So, T(n)≥cn2 for c=c1 and n0=1. Therefore, T(n) is in Ω(n2) by the definition.

It is also true that the equation of the example above is in Ω(n).